



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---------------------|-------------|----------------------|---------------------|------------------|
| 10/787,520 | 02/26/2004 | Kent F. Hayes JR. | RSW920030232US1 | 6473 |
| 75532 | 7590 | 07/26/2011 | | |
| LEE LAW, PLLC | | | EXAMINER | |
| IBM CUSTOMER NUMBER | | | KEEHN, RICHARD G | |
| P.O. BOX 189 | | | | |
| PITTSBORO, NC 27312 | | | ART UNIT | PAPER NUMBER |
| | | | 2456 | |
| | | | MAIL DATE | DELIVERY MODE |
| | | | 07/26/2011 | PAPER |

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/787,520
Filing Date: February 26, 2004
Appellant(s): HAYES, KENT F.

Christopher B. Lee (58793)
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 5/02/2011 appealing from the Office action mailed 12/06/2010.

(1) Real Party in Interest

The examiner has no comment on the statement, or lack of statement, identifying by name the real party in interest in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The following is a list of claims that are rejected and pending in the application:

1, 3-16, 18-25 and 27-32.

(4) Status of Amendments After Final

The examiner has no comment on the appellant's statement of the status of amendments after final rejection contained in the brief.

(5) Summary of Claimed Subject Matter

The examiner has no comment on the summary of claimed subject matter contained in the brief.

(6) Grounds of Rejection to be Reviewed on Appeal

The examiner has no comment on the appellant's statement of the grounds of rejection to be reviewed on appeal. Every ground of rejection set forth in the Office action from which the appeal is taken (as modified by any advisory actions) is being maintained by the examiner except for the grounds of rejection (if any) listed under the

Art Unit: 2456

subheading "WITHDRAWN REJECTIONS." New grounds of rejection (if any) are provided under the subheading "NEW GROUNDS OF REJECTION."

NEW GROUND(S) OF REJECTION

Not Applicable.

WITHDRAWN REJECTIONS

The following grounds of rejection are not presented for review on appeal because they have been withdrawn by the examiner. Not Applicable.

(7) Claims Appendix

The examiner has no comment on the copy of the appealed claims contained in the Appendix to the appellant's brief.

(8) Evidence Relied Upon

| | | |
|-----------------|-----------------|---------|
| 2003/0023661 A1 | Clohessy et al. | 1-2003 |
| 6,493,871 B1 | McGuire et al. | 12-2002 |
| 2003/0131226 A1 | Spencer et al. | 7-2003 |
| 2005/0004974 A1 | Sharma et al. | 1-2005 |

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

1. Claims 25 and 27-32 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

As to Claims 25 and 27-32, Applicant has provided evidence that applicant intends the invention to be embodied as a program product stored on a storage medium. The broadest reasonable interpretation of a claim drawn to storage

Art Unit: 2456

medium typically covers forms of non-transitory tangible media and transitory propagating signals per se. As such, the claims are drawn to a form of energy.

Energy is not one of the four categories of invention and therefore claims 25 and 27-32 are not statutory. Energy is not a series of steps and thus is not a process. Energy is not a physical article or object and such is not a machine or manufacture. Energy is not a combination of substances and therefore not a composition of matter.

2. Claims 1, 3, 5-11, 13-16, 18, 20-23, 25, 27 and 29-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 2003/0023661 A1 (Clohessy et al.), and further in view of US 6,493,871 B1 (McGuire et al.).

As to Claims 1, 10, 16 and 25, Clohessy et al. disclose a computer-implemented method, a computer-implemented method, a computerized system for resolving prerequisites, and a program product stored on a storage medium and executed by a computer, respectively, for client devices in an Open Service Gateway Initiative (OSGi) framework, comprising:

{Claim 16 ONLY} a memory that stores OSGi bundle information at a server device (Clohessy et al. disclose the memory - ¶ [0022]);

{Claim 16 ONLY} a processor programmed to execute (Clohessy et al. disclose the processor - ¶ [0022]);

determining, on a server device, an OSGi bundle to be loaded on a client device (Clohessy et al. discloses determining by the server, the runtime resources needed on,

Art Unit: 2456

and to be loaded on the client device - Page 4, ¶ [0039]; ¶ [0038] recites the use of OSGi bundles) the prerequisites comprising a set of all that are necessary for utilizing the OSGi bundle (Clohessy et al. discloses determining by the server, the runtime resources needed on, and to be loaded on the client device - Page 4, ¶ [0039]; ¶¶ [0035 and 0038] recite the use of OSGi bundles including amount of RAM, the number of threads, and the number of sockets for the associated application component and that these are service resources as part of the open Service Gateway initiative; ¶ [0035] also recites that the RDL is comprised of Java which is package based and provides interface components. Figure 2A discloses that the application is run on a PDA which discloses the application interface as well);

OSGi bundles (Clohessy et al. discloses determining by the server, the runtime resources needed on, and to be loaded on the client device - Page 4, ¶ [0039]; ¶ [0038] recites the use of OSGi bundles);

identifying a final set OSGi bundles on the server device that fulfills the resource limitations of the client device (Clohessy et al. discloses that the bundles needed are identified, and not sent until the client has sufficient resources available - Page 4, ¶ [0043]); and

would not require more client device OSGi package and OSGi service interface resources than the current OSGi package and OSGi service interface resources of the client device (Clohessy et al. disclose determining the client resources and whether the client has sufficient resources for the bundles - ¶ [0036]; ¶¶ [0035 and 0038] recite the use of OSGi bundles including amount of RAM, the number of threads, and the number

Art Unit: 2456

of sockets for the associated application component and that these are service resources as part of the open Service Gateway initiative; ¶ [0035] also recites that the RDL is comprised of Java which is package based and provides interface components. Figure 2A discloses that the application is run on a PDA which discloses the application interface as well) and

would require more client device OSGi package and OSGi service interface resources than the current OSGi package and OSGi service interface resources of the client device (Clohessy et al. disclose determining the client resources and whether the client has sufficient resources for the bundles - ¶ [0036]; ¶¶ [0035 and 0038] recite the use of OSGi bundles including amount of RAM, the number of threads, and the number of sockets for the associated application component and that these are service resources as part of the open Service Gateway initiative; ¶ [0035] also recites that the RDL is comprised of Java which is package based and provides interface components. Figure 2A discloses that the application is run on a PDA which discloses the application interface as well); and

automatically recursively resolving the prerequisites (Clohessy et al. – Figure 4 shows the recursive path used to resolve prerequisites, 104-106-108-109-110-112-104-etc. until 114 or END; and ¶ [0038]).

Clohessy et al. do not explicitly communicating, prior to communicating any of the {software} to the client device, a list of the prerequisites from the server to the client device; receiving a response from the client device, wherein the response identifies any resource limitations of the client device determined by the client device based on a

Art Unit: 2456

comparison of the list of the prerequisites and current resources of the client device, the resource limitations comprising all prerequisites of the list of the prerequisites that are not currently present on the client device; resolving, upon determining that the list of the prerequisites that are not currently present on the client device, the prerequisites by identifying a final set {of software to be sent to the client} on the server that fulfills the prerequisites within the resource limitations of the client device; and substituting, upon determining that the list of the prerequisites that are not currently present on the client device, at least one other {software} bundle that operates within the resource limitations of the client device for one of the {software} bundles and one of the prerequisites of the list of the prerequisites that are not currently present on the client device. However McGuire et al. disclose

communicating, prior to communicating any of the {software} to the client device, a list of the prerequisites from the server device to the client device (McGuire et al. disclose communicating the list from the server to the client – Column 4, lines 17-21);

receiving a response from the client device, wherein the response identifies any resource limitations of the client device determined by the client device based on a comparison of the list of the prerequisites and current resources of the client device, the resource limitations comprising all prerequisites of the list of the prerequisites that are not currently present on the client device (McGuire et al. disclose the client executing the comparison of its resources against prerequisites and sending the needed list to the server - Column 4, lines 21-27); and

Art Unit: 2456

resolving via the server device, in response to determining that the list of the prerequisites that are not currently present on the client device, the prerequisites by identifying a final set {of software to be sent to the client} on the server that fulfills the prerequisites within the resource limitations of the client device (McGuire et al. disclose the server sending the necessary files to the client based on a response from the client on resource deficiency - Column 4, lines 30-37); and

substituting via the server device, in response to determining that the list of the prerequisites that are not currently present on the client device, at least one other {software} bundle that operates within the resource limitations of the client device for one of the {software} bundles and one of the prerequisites of the list of the prerequisites that are not currently present on the client device (McGuire et al. disclose the substitution of prerequisites - Column 4, lines 32-33; and the server sending the necessary files to the client based on a response from the client on resource deficiency - Column 4, lines 30-37); and

{CLAIM 10 ONLY}

caching information derived from the response on the server (McGuire et al. discloses server caching – Column 13, lines 35-38).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the process of resolving client prerequisites taught by McGuire et al., with sending OSGi bundles from a server to a client taught by Clohessy et al., in order to minimize the amount of data to be downloaded by downloading only

Art Unit: 2456

those files needed to update the client computer and that will fulfill the client's needs (McGuire et al. – Column 4, lines 13-17 and Abstract).

As to Claim 3, the combination of Clohessy et al. and McGuire et al. discloses the method of claim 1, further comprising loading the final set of OSGi bundles on the client device if the prerequisites are completely resolved (Clohessy et al. – Figure 4, elements 108 and 114 show loading the final set of OSGi bundles on the client device).

As to Claim 5, the combination of Clohessy et al. and McGuire et al. discloses the method of claim 1, wherein the prerequisites comprise at least one item selected from a group consisting of a service, a package and a computer resource needed by client device (Clohessy et al. - Page 2, paragraphs 0024 and 0025 disclose the prerequisite resources being flash memory and RAM, which are at least one item selected from the group consisting of a service, a package and a computer resource, i.e. computer resources).

As to Claim 6, the combination of Clohessy et al. and McGuire et al. discloses the method of claim 1, further comprising caching information derived from the response on the server device (McGuire et al. discloses server caching – Column 13, lines 35-38).

The motivation and obviousness arguments are the same as in Claim 1.

As to Claim 7, the combination of Clohessy et al. and McGuire et al. discloses the method of claim 1, wherein the method is applied in the presence of a low bandwidth or high cost connection between the server device and the client device (Clohessy et al. – Page 1, paragraph 0002 discloses that the client device has limited system resources which would include bandwidth related resources such as threads, sockets, memory, RAM, etc. Paragraph 0003 further recites that the client device requires frequent loading and unloading due to the narrow bandwidth as compared to a desktop PC).

As to Claim 8, the combination of Clohessy et al. and McGuire et al. discloses the method of claim 1, wherein the final set of OSGi bundles include OSGi bundles that are identified from a repository accessed by the server device (Clohessy et al. – Page 4, paragraphs 0042 and 0043 disclose the process of loading OSGi bundles residing on the server to the client).

As to Claim 9, the combination of Clohessy et al. and McGuire et al. discloses the method of claim 1, further comprising:

receiving the prerequisites on the client device (McGuire et al. disclose the client executing the comparison of its resources against prerequisites and sending the needed list to the server - Column 4, lines 21-27);

determining whether the client device has the prerequisites, wherein any of the prerequisites that the client device does not have represent the resource limitations (McGuire et al. disclose the client executing the comparison of its resources against prerequisites and sending the needed list to the server - Column 4, lines 21-27); and sending the response to the server device, wherein the response includes the resource limitations (McGuire et al. disclose the client executing the comparison of its resources against prerequisites and sending the needed list to the server - Column 4, lines 21-27).

The motivation and obviousness arguments are the same as in Claim 1.

As to Claim 11, the combination of Clohessy et al. and McGuire et al. discloses the method of claim 10, further comprising loading the final set of OSGi bundles on the client device if the prerequisites are completely resolved (Clohessy et al. – Figure 4, elements 108 and 114 show loading the final set of OSGi bundles on the client device).

As to Claim 13, the combination of Clohessy et al. and McGuire et al. discloses the method of claim 10, wherein the prerequisites comprise at least one item selected from a group consisting of a service, a package and a computer resource needed by client device (Clohessy et al. - Page 2, paragraphs 0024 and 0025 disclose the prerequisite resources being flash memory and RAM, which are at least one item selected from the group consisting of a service, a package and a computer resource, i.e. computer resources).

As to Claim 14, the combination of Clohessy et al. and McGuire et al. discloses the method of claim 10, wherein the method is applied in the presence of a low bandwidth or high cost connection between the server device and the client device (Clohessy et al. – Page 1, paragraph 0002 recites that the client device has limited system resources which would include bandwidth related resources such as threads, sockets, memory, RAM, etc. Paragraph 0003 further recites that the client device requires frequent loading and unloading due to the narrow bandwidth as compared to a desktop PC).

As to Claim 15, the combination of Clohessy et al. and McGuire et al. discloses the method of claim 10, further comprising:

receiving the prerequisites on the client device (McGuire et al. disclose the client executing the comparison of its resources against prerequisites and sending the needed list to the server - Column 4, lines 21-27);

determining whether the client device has the prerequisites, wherein any of the prerequisites that the client device does not have represent the resource limitations (McGuire et al. disclose the client executing the comparison of its resources against prerequisites and sending the needed list to the server - Column 4, lines 21-27); and

sending the response to the server device , wherein the response includes the resource limitations (McGuire et al. disclose the client executing the comparison of its

Art Unit: 2456

resources against prerequisites and sending the needed list to the server - Column 4, lines 21-27).

The motivation and obviousness arguments are the same as in Claim 1.

As to Claim 18, the combination of Clohessy et al. and McGuire et al. discloses the system of claim 16, where the processor is further programmed to execute a bundle loading system for loading the final set of OSGi bundles on the client device if the prerequisites are completely resolved (Clohessy et al. – Figure 4, elements 108 and 114 show loading the final set of OSGi bundles on the client device).

As to Claim 20, the combination of Clohessy et al. and McGuire et al. discloses the system of claim 16, wherein the prerequisites comprise at least one item selected from a group consisting of a service, a package and a computer resource needed by client device (Clohessy et al. - Page 2, paragraphs 0024 and 0025 recite the prerequisite resources being flash memory and RAM, which are at least one item selected from the group consisting of a service, a package and a computer resource, i.e. computer resources).

As to Claim 21, the combination of Clohessy et al. and McGuire et al. discloses the system of claim 16, where the processor is further programmed to execute a response caching system for caching information derived from the response within the

Art Unit: 2456

memory at the server device (McGuire et al. discloses server caching – Column 13, lines 35-38).

The motivation and obviousness arguments are the same as in Claim 1.

As to Claim 22, the combination of Clohessy et al. and McGuire et al. discloses the system of claim 16, where the memory comprises a repository and wherein the final set of OSGi bundles includes OSGi bundles that are identified from the repository accessed by the server device (Clohessy et al. – Page 4, paragraphs 0042 and 0043 disclose the process of loading OSGi bundles residing on the server to the client).

As to Claim 23, the combination of Clohessy et al. and McGuire et al. discloses the system of claim 16, where the processor is further programmed to process the response generated via:

an analysis system executing on the client device that determines whether the client device has the prerequisites, wherein any prerequisites that the client device does not have are identified as the resource limitations (McGuire et al. disclose the client executing the comparison of its resources against prerequisites and sending the needed list to the server - Column 4, lines 21-27); and

a response system that sends the response from the client device to the server device (McGuire et al. discloses the server sending the necessary files to the client based on a response from the client on resource deficiency - Column 4, lines 30-37).

The motivation and obviousness arguments are the same as in Claim 1.

As to Claim 27, the combination of Clohessy et al. and McGuire et al. discloses the program product of claim 25, further comprising program code for loading the final set of OSGi bundles on the client device if the prerequisites are completely resolved (Clohessy et al. – Figure 4, elements 108 and 114 show loading the final set of OSGi bundles on the client device).

As to Claim 29, the combination of Clohessy et al. and McGuire et al. discloses the program product of claim 25, wherein the prerequisites comprise at least one item selected from a group consisting of a service, a package and a computer resource needed by client device (Clohessy et al. - Page 2, paragraphs 0024 and 0025 recite the prerequisite resources being flash memory and RAM, which are at least one item selected from the group consisting of a service, a package and a computer resource, i.e. computer resources).

As to Claim 30, the combination of Clohessy et al. and McGuire et al. discloses the program product of claim 25, further comprising program code for caching the information derived from the response on the server device (Mcguire et al. discloses server caching – Column 13, lines 35-38).

As to Claim 31, the combination of Clohessy et al. and McGuire et al. discloses the program product of claim 25, wherein the final set of OSGi bundles includes OSGi

bundles that are identified from a repository accessed by the server device (Clohessy et al. – Page 4, paragraphs 0042 and 0043 recite the process of loading OSGi bundles residing on the server to the client).

As to Claim 32, the combination of Clohessy et al. and McGuire et al. discloses the program product of claim 25, further comprising:

program code for determining whether the client device has the prerequisites, wherein any prerequisites that the client device does not have are identified as the resource limitations (McGuire et al. disclose the client executing the comparison of its resources against prerequisites and sending the needed list to the server - Column 4, lines 21-27); and

program code for sending the response from the client device to the server device (McGuire et al. discloses the server sending the necessary files to the client based on a response from the client on resource deficiency - Column 4, lines 30-37).

The motivation and obviousness arguments are the same as in Claim 1.

3. Claims 4, 12, 19 and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over the combination of Clohessy et al. and McGuire et al., and further in view of US 2003/0131226 A1 (Spencer et al.).

As to Claim 4, the combination of Clohessy et al. and McGuire et al. disclose the method of claim 3, wherein the loading comprises the server device instructing the client

device to load the final set of OSGi bundles (Clohessy et al. – Figure 4, elements 108 and 114 show loading the final set of OSGi bundles on the client device) .

The combination of Clohessy et al. and McGuire et al. do not disclose in a particular order, but Spencer et al. disclose in a particular order (Spencer et al. – Page 1, paragraph 0004 discloses downloading components in a particular order).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the method of downloading in a particular order taught by Spencer et al., with the server instructing the client device to load the final set of OSGi bundles taught by the combination of Clohessy et al. and McGuire et al., in order to load components according to whether they that require the presence of others on the user's device (Spencer et al. – Page 1, paragraph 0004).

As to Claim 12, the combination of Clohessy et al. and McGuire disclose the method of claim 11, wherein the loading comprises the server device instructing the client device to load the final set of OSGi bundles (Clohessy et al. – Figure 4, elements 108 and 114 show loading the final set of OSGi bundles on the client device).

The combination of Clohessy et al. and McGuire et al. does not disclose in a particular order, but Spencer et al. disclose in a particular order (Spencer et al. – Page 1, paragraph 0004 recites downloading components in a particular order).

The motivation and obviousness arguments are the same as in Claim 4.

As to Claim 19, the combination of Clohessy et al. and McGuire et al. disclose the system of claim 18, wherein the bundle loading system comprises an instruction passing system for instructing the client device to load the final set of OSGi bundles (Clohessy et al. – Figure 4, elements 108 and 114 show loading the final set of OSGi bundles on the client device).

The combination of Clohessy et al. and McGuire et al. do not disclose in a particular order, but Spencer et al. disclose in a particular order (Spencer et al. – Page 1, paragraph 0004 recites downloading components in a particular order).

The motivation and obviousness arguments are the same as in Claim 4.

As to Claim 28, the combination of Clohessy et al. and McGuire et al. disclose the program product of claim 27, wherein the program code for loading comprises program code for instructing the client device to load the final set of OSGi bundles (Clohessy et al. – Figure 4, elements 108 and 114 show loading the final set of OSGi bundles on the client device).

The combination of Clohessy et al. and McGuire et al. do not disclose in a particular order, but Spencer et al. disclose in a particular order (Spencer et al. – Page 1, paragraph 0004 recites downloading components in a particular order).

The motivation and obviousness arguments are the same as in Claim 4.

4. Claim 24 is rejected under 35 U.S.C. 103(a) as being unpatentable over the combination of Clohessy et al. and McGuire et al. as applied to claim 16 above, and further in view of US 2005/0004974 A1 (Sharma et al.).

As to Claim 24, the combination of Clohessy et al. and McGuire et al. disclose the system of claim 16.

The combination of Clohessy et al. and McGuire et al. do not disclose wherein the system uses SyncML DM protocol for communication between the client device and the server device, but Sharma et al. disclose wherein the system uses SyncML DM protocol for communication between the client device and the server device (Sharma et al. – Page 9, paragraphs 0097 and 0099 recite the use of SyncML Device Management and OSGi to communicate between client and server).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the use of SyncML Device Management taught by Sharma et al., with the communication between client and server taught by the combination of Clohessy et al. and McGuire et al.

One of ordinary skill in the art at the time the invention was made would have been motivated to explicitly support the ability to change service settings on a mobile device and to be able to download services to it (Sharma et al. – Page 9, paragraph 0099).

(10) Response to Argument

1. The combination of Clohessy et al. and McGuire et al. discloses Claims 1, 3, 5-11, 13-16, 18, 20-23, 25, 27 and 29-32.

Appellant's arguments start on page 17 of the Brief.

Argument A: (Brief at 17)

Appellant argues that "determining currently available runtime resources is very different from determining the prerequisites themselves for an OSGi bundle." (Brief at 17). Appellant argues that Clohessy only "discloses a single determination of prerequisites...and does not disclose returning to re-determine the prerequisites." (Brief at 17).

Response to Argument A:

These arguments are not claimed. The claim language requires determining a set of OSGi bundles, then automatically recursively resolving the set based on the prerequisites not currently on the client device. In other words, the prerequisites needed for the client to run its application do not change, but the prerequisites needed to be sent to the client will change based on what the client has available and does not have available. This is what the cited combination of references discloses. (Clohessy at ¶¶[0035-0039] and Figure 4)

Argument B: (Brief at 17)

Appellant continues by arguing that McGuire was cited to disclose "substituting via the server device, ... at least one other OSGi bundle that operates within the resource limitations of the client device." (Brief at 17).

Response to Argument B:

As an initial matter, Examiner did not use McGuire to disclose OSGi bundles, so Appellant has misspoken. Nonetheless, Clohessy clearly discloses OSGi prerequisites. (Clohessy at ¶¶ [0038-0039]). The argument continues that McGuire's setup program runs on the client computer. (Brief at 17). It is not clear based on this paragraph extending into page 18 exactly what Appellant is arguing, but Examiner reasonably interprets the argument to be that because the setup program is running on the client computer, the server computer is not doing the substituting as required in the claim language. However, Clohessy clearly discloses the server's automatic resolution of prerequisites in Figure 4, elements 104-112. This resolution is performed on the server and is automatically executed on and by the server. Appellant's perceived argument that because McGuire has a user interface at the client used for prerequisite resolution, the process is not automatic is not persuasive because even if the client is operated with human intervention, the server's process is still automatic. The server may have to wait for input from the client device, the delay caused by human intervention, but the server's process is automatic. The claims require server automatic process, not server and client automatic processes. In other words, Appellant is doing an apples-to-oranges comparison. Appellant's argument is that the *client* is not automatic, and tries to equate the *client's* mode of operation to the *server*.

Argument C: (Brief at 18)

Appellant continues by arguing that the combination of Clohessy and McGuire would render Clohessy unsatisfactory for its intended purpose. (Brief at 18).

Response to Argument C:

However, looking at Clohessy's Figure 4, element 104, it is clear that a reference such as McGuire that further describes a process to resolve client prerequisites by describing client-side operation fits perfectly into Clohessy's Figure 4, element 104. Examiner stated in the final Office action on pages 8 and 9 that it would have been obvious to combine the references in order to minimize the amount of data to be downloaded by downloading only those files needed to update the client computer and that will fulfill the client's needs. This is a perfect supplement to Clohessy, and does not render Clohessy inoperative.

Appellant appears to argue that sending the setup program to the client device changes the fundamental principle of McGuire. How? Sending the setup program IS in McGuire at 4:17-21. You can't change the fundamental principle of a reference by including something already present IN the reference. Furthermore, McGuire is the secondary reference. Appellant argues that the disclosed client in McGuire is not a server. As stated above, Clohessy discloses the server automatic process in Figure 4.

Argument D: (Brief at 19)

Appellant argues that the "Mcguire reference further teaches away from Appellant's claims by disclosing client-side processing for determination of the needed files instead of server-side processing as claimed". (Brief at 19).

Response to Argument D:

This argument looks only at one reference instead of the combination. As stated above, Clohessy clearly discloses the server's automatic resolution of prerequisites in Figure 4, elements 104-112. This resolution is performed on the server and is

Art Unit: 2456

automatically executed on and by the server. Appellant's perceived argument that because McGuire has a user interface at the client used for prerequisite resolution, the process is not automatic is not persuasive because even if the client is operated with human intervention, the server's process is still automatic. The server may have to wait for input from the client device, the delay caused by human intervention, but the server's process is automatic. The claims require server automatic process, not server and client automatic processes. In other words, Appellant is doing an apples-to-oranges comparison. Appellant's argument is that the *client* is not automatic, and tries to equate the *client's* mode of operation to the *server*.

Argument E: (Brief at 19-23)

Appellant provides a summary of the references cited on pages 19-23.

Response to Argument E:

Examiner's response to Appellant's characterization of the cited references is included throughout this section in response to Appellant's arguments. Appellant resumes argument on page 26 of the Appeal Brief.

Argument F: (Brief at 26)

Appellant argues that "[A]s a preliminary matter, the Patent Office admits that the Clohessy reference does not disclose resolving, in response to determining that the list of the prerequisites that are not currently present on the client device...,the prerequisites by identifying a final set...on the server device that fulfills the prerequisites within the resource limitations of the client device." Appellant cites pages 7-8 of the Office action dated 12/6/2010.

Response to Argument F:

However, if you read those pages, you'll see that Appellant has misspoken again. The Final Office action recites "Clohessy et al. do not explicitly communicating, prior to communicating any of the {software} to the client device, a list of the prerequisites from the server to the client device; receiving a response from the client device, wherein the response identifies any resource limitations of the client device determined by the client device based on a comparison of the list of the prerequisites and current resources of the client device, the resource limitations comprising all prerequisites of the list of the prerequisites that are not currently present on the client device; resolving, upon determining that the list of the prerequisites that are not currently present on the client device, the prerequisites by identifying a final set {of software to be sent to the client} on the server that fulfills the prerequisites within the resource limitations of the client device; and substituting, upon determining that the list of the prerequisites that are not currently present on the client device, at least one other {software} bundle that operates within the resource limitations of the client device for one of the {software} bundles and one of the prerequisites of the list of the prerequisites that are not currently present on the client device."

Argument G: (Brief at 28, 29)

Appellant argues that the Clohessy Figure 4 reference returns to step 104, not step 100 and therefore does not recursively resolve the prerequisites. (Brief at 28, 29).

Response to Argument G:

As Appellant admits, Clohessy determines an initial list of prerequisites needed for the client to run its application. Clohessy then automatically and recursively resolves the prerequisites the client actually needs by comparing what the list requires vs. what the client has available to it. Appellant has misinterpreted his claim requirements. Looking at Claim 1, first limitation, the server determines prerequisites for an OSGi bundle to be loaded on a client device, the same as Clohessy Figure 4, elements 100-102. However, the fourth and fifth limitations clearly address resolving a final set of prerequisites based on what the client has available to it, just as in Clohessy, Figure 4, elements 104-112. Both Appellant's claimed invention and the Clohessy reference create an initial set of prerequisites, and both revise based on what the client needs at the moment. There simply is no difference.

Argument H: (Brief at 29)

Appellant's arguments on page 29 are repeated from those discussed earlier.

Response to Argument H:

Appellant's arguments on page 29 are repeated from those discussed earlier.

Argument I: (Brief at 30)

Appellant argues that the Patent Office has unreasonable over-broadened the claim language by "citing disclosure of a repeating process for removal of applications from a portable device and re-determining currently available runtime system resources (which are different from prerequisites both as claimed by Appellant and as disclosed by the Clohessy reference) against Appellant's claimed automatically recursively resolving via the server device the prerequisites..." (Brief at 30).

Response to Argument I:

However, memory is one of the prerequisites needed to run a client application, and recursive removal of applications to free up memory is certainly within the definition of recursively resolving prerequisites. Appellant's claims do not preclude removal of applications to resolve prerequisites. Furthermore, Clohessy clearly indicates that the resolution of prerequisites can be performed by either the client or the server (Clohessy - ¶ [0043 last line]).

Argument J: (Brief at 30)

Appellant's arguments that "a person of skill in the art would not be motivated..." is based on Appellant's own hypothetical reasoning regarding processing time and development time, but without any evidence to support it. (Brief at 30).

Response to Argument J:

Appellant offers no evidence to support this argument.

Argument K: (Brief at 31)

Appellant argues that "Appellant defines that recursive resolution is directed to situations where prerequisites themselves have prerequisites and repeatedly resolving these prerequisites of prerequisites recursively until all prerequisites are resolved or until no other combinations of bundles remain ... bundles 20 could themselves have prerequisites".

Response to Argument K:

This argument is unpersuasive for two reasons:

First, Appellant's alleged support that "bundles 20 **could** themselves have prerequisites" {**emphasis** by Examiner} clearly indicates that prerequisites may also *not have prerequisites*. So allegedly there are two types of prerequisites: those that have prerequisites themselves, and those that don't. The claims only require resolution of prerequisites, not resolution of prerequisites that have prerequisites themselves. Therefore Appellant argues that which has not been claimed. Second, Appellant impermissibly demands that the Patent Office read the specification in to the claims, by seeking to limit the definition of "prerequisite" to those which require additional prerequisites. This whole argued notion of "recursion" meaning the resolution of prerequisites that require more prerequisites is not claimed nor supported in the disclosure. Therefore, the Patent Office has not unreasonably broadened the claim language, but rather the Appellant unreasonably narrowed the claim scope to try to overcome a reference by arguing that which is not supported in the disclosure.

Argument L: (Brief at 31)

Appellant argues that even if Clohessy's Figure 4 returned back to step 100 instead of 104, prerequisites would be the same prerequisites determined the first time without any change.

Response to Argument L:

This argument is based on Appellant's (not Examiner's) change to the interpretation of Clohessy's Figure 4. Nonetheless, as discussed above, Clohessy determines an initial list of prerequisites needed for the client to run its application. Clohessy then automatically and recursively resolves the prerequisites the client

Art Unit: 2456

actually needs by comparing what the list requires vs. what the client has available to it. Appellant has misinterpreted his claim requirements. Looking at Claim 1, first limitation, the server determines prerequisites for an OSGi bundle to be loaded on a client device, the same as Clohessy Figure 4, elements 100-102. However, the fourth and fifth limitations clearly address resolving a final set of prerequisites based on what the client has available to it, just as in Clohessy, Figure 4, elements 104-112. Both Appellant's claimed invention and the Clohessy reference create an initial set of prerequisites, and both revise based on what the client needs at the moment. There simply is no difference.

Argument M: (Brief at 32, 33)

Appellant's arguments on page 32 and the top of page 33 repeat arguments discussed above.

Response to Argument M:

Appellant's arguments on page 32 and the top of page 33 repeat arguments discussed above.

Argument N: (Brief at 33)

Appellant argues that the claimed server process differs from that using a manual client-side interface as disclosed in the cited art. (Brief at 33).

Response to Argument N:

The claims do not preclude client side manual or semi-automatic intervention. Indeed, Claim 1's third limitation recites "receiving a response from the client device." The claim language does not preclude client-side human intervention.

Argument O: (Brief at 34)

Appellant argues that the cited references do not disclose “in response to determining that the list ...”

Response to Argument O:

Clearly, the loop from element 112 back to element 104 in Clohessy’s Figure 4 disclose “in response to”. There is closed-loop feedback until all prerequisites are met.

Argument P: (Brief at 35-39)

Appellant’s arguments on pages 35-39(top) repeat those discussed above.

Response to Argument P:

Appellant’s arguments on pages 35-39(top) repeat those discussed above.

Argument Q: (Brief at 39-42)

Appellant argues that McGuire’s files are not OSGi bundles. (Brief at 39-42).

Response to Argument Q:

However, McGuire was not relied upon to disclose OSGi bundles. In addition, Appellant is arguing against a single reference instead of the combination cited.

Argument R: (Brief at 43)

Appellant argues that McGuire would change two fundamental intended purposes of Clohessy. The first being the return to step 100. (Brief at 43).

Response to Argument R:

Clearly this change in Clohessy's Figure 4 is Appellant's own proposed change. First, the patent Office did not propose this change, so the argument is moot. Second, changing runtime resource calculation is nothing more than resolving the resources

Art Unit: 2456

needed on the client based on what the client currently has or has not. In addition, step 100 in Clohessy's Figure 4 is the initial prerequisites needed to run the client's application, not a revised list based on the client's resources. There simply is no reason to return to step 100. As discussed above, looking at Clohessy's Figure 4, element 104, it is clear that a reference such as McGuire that further describes a process to resolve client prerequisites by describing client-side operation fits perfectly into Clohessy's Figure 4, element 104. Examiner stated in the final Office action on pages 8 and 9 that it would have been obvious to combine the references in order to minimize the amount of data to be downloaded by downloading only those files needed to update the client computer and that will fulfill the client's needs. This is a perfect supplement to Clohessy, and does not render Clohessy inoperative. Clohessy determines an initial list of prerequisites needed for the client to run its application. Clohessy then automatically and recursively resolves the prerequisites the client actually needs by comparing what the list requires vs. what the client has available to it. Appellant has misinterpreted his claim requirements. Looking at Claim 1, first limitation, the server determines prerequisites for an OSGi bundle to be loaded on a client device, the same as Clohessy Figure 4, elements 100-102. However, the fourth and fifth limitations clearly address resolving a final set of prerequisites based on what the client has available to it, just as in Clohessy, Figure 4, elements 104-112. Both Appellant's claimed invention and the Clohessy reference create an initial set of prerequisites, and both revise based on what the client needs at the moment. There simply is no difference.

Argument S: (Brief at 44, 45)

Appellant argues that McGuire teaches away from Clohessy because of the client interface. (Brief at 44, 45).

Response to Argument S:

As mentioned above, the claims do not preclude client-side human interface, and the client-side interface is a natural part of Clohessy's Figure 4, step 104.

Argument T: (Brief at 46, 47)

Appellant's arguments in section 4, pages 45-46 repeat those discussed above.

Response to Argument T:

Appellant's arguments in section 4, pages 45-46 repeat those discussed above.

Argument U: (Brief at 46, 47)

Appellant argues impermissible hindsight and picking and choosing. (Brief at 46, 47).

Response to Argument U:

Looking at Clohessy's Figure 4, element 104, it is clear that a reference such as McGuire that further describes a process to resolve client prerequisites by describing client-side operation fits perfectly into Clohessy's Figure 4, element 104. Examiner stated in the final Office action on pages 8 and 9 that it would have been obvious to combine the references in order to minimize the amount of data to be downloaded by downloading only those files needed to update the client computer and that will fulfill the client's needs. This is a perfect supplement to Clohessy, and does not render Clohessy inoperative, nor change its intended purpose. In response to applicant's argument that the examiner's conclusion of obviousness is based upon improper hindsight reasoning,

Art Unit: 2456

it must be recognized that any judgment on obviousness is in a sense necessarily a reconstruction based upon hindsight reasoning. But so long as it takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper. See *In re McLaughlin*, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971). In response to applicant's argument that Examiner picked and chose elements of McGuire, the test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference; nor is it that the claimed invention must be expressly suggested in any one or all of the references. Rather, the test is what the combined teachings of the references would have suggested to those of ordinary skill in the art. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981).

Argument V: (Brief at 48)

Appellant argues against the rejection of dependant claims in section 6. The arguments are based on arguments discussed above. Appellant also argues that "by identifying a final set of OSGi bundles" is not disclosed. (Brief at 48).

Response to Argument V:

However, as discussed above, the final set is resolved by the recursive process in Clohessy's Figure 4, elements 104-112.

Argument W: (Brief in Section 7)

Appellant argues against the rejection of dependant claims in section 7.

Response to Argument W:

The arguments are based on arguments discussed above. As to Appellant's RAM argument at page 49, RAM is a prerequisite for running client applications hence fits the description. Examiner points Appellant to Appellant's Specification, ¶ [0003], where RAM is clearly disclosed as a computer resource needed by the client.

Argument X: (Brief at 50)

Appellant argues that the caching in the cited art is not the same as that claimed, based on Appellant's "belief". (Brief at 50).

Response to Argument X:

Claim 1 requires "caching information derived from the response on the server", which McGuire discloses at 13:35-38. Clearly the server is using caching for data transfer in and out of the server, therefore necessarily covers responses from the server.

Argument Y: (Brief at 51)

Appellant argues that Clohessy does not disclose the limitation of Claim 7. (Brief at 51).

Response to Argument Y:

Clohessy et al. – ¶ [0002] discloses that the client device has limited system resources which would include bandwidth related resources such as threads, sockets, memory, RAM, etc. ¶ [0003] further recites that the client device requires frequent loading and unloading due to the narrow bandwidth as compared to a desktop PC. Appellant continues to allege missing elements based on Appellant's "belief."

Argument Z: (Brief at 52, 53)

Appellant argues that a repository is not disclosed as claimed. (Brief at 52, 53).

Response to Argument Z:

Clohessy et al. – Page 4, paragraphs 0042 and 0043 disclose the process of loading OSGi bundles residing on the server to the client.

Argument AA: (Brief at 53, 54)

Appellant argues against the rejection of dependant claims based on independent claim arguments and the theory of McGuire not disclosing OSGi bundles. (Brief at 53, 54)

Response to Argument AA:

As discussed above, McGuire was not relied upon to disclose OSGi bundles.

2. The combination of Clohessy et al., McGuire et al. and Spencer et al. disclose Claims 4, 12, 19 and 28.

Argument BB: (Brief at 54)

Appellant's arguments on page 54 are based on previously discussed arguments.

Response to Argument BB:

Appellant's arguments on page 54 are based on previously discussed arguments.

Argument CC: (Brief at 55)

Appellant argues that there is no specific instruction from the server to the client device and cites ¶ [0046] in support. (Brief at 55).

Response to Argument CC:

However, Clohessy's ¶ [0043] makes it crystal clear that "the identified one or more new application components will be loaded into the portable device at step 114," (referring to Figure 4) and "the component manager in the base system 40 (FIG. 3) of the portable device makes the decision in step 108. *In an alternative embodiment, the server through which the one or more new application components will be delivered makes the decision.*" {emphasis by Examiner}

3. The combination of Clohessy et al., McGuire et al. and Sharma et al. discloses Claim 24.

Argument DD: (Brief at 57)

Appellant argues against the SyncML disclosure of Sharma. (Brief at 57).

Response to Argument DD:

Examiner points out that the claim language "wherein the system uses SyncML DM protocol for communication between the client device and the server device" is disclosed by Sharma et al. – Page 9, paragraphs 0097 and 0099 recite the use of SyncML Device Management and OSGi to communicate between client and server. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the use of SyncML Device Management taught by Sharma et al., with the communication between client and server taught by the combination of Clohessy et al. and McGuire et al. One of ordinary skill in the art at the time the invention was made would have been motivated to explicitly support the ability to change service settings on

Art Unit: 2456

a mobile device and to be able to download services to it (Sharma et al. – Page 9, paragraph 0099).

4. Claims 25 and 27-32 are properly rejected under 35 U.S.C. 101.

As to Claims 25 and 27-32, Appellant has provided evidence that applicant intends the invention to be embodied as a program product stored on a storage medium. The broadest reasonable interpretation of a claim drawn to storage medium typically covers forms of non-transitory tangible media and transitory propagating signals per se. As such, the claims are drawn to a form of energy. Energy is not one of the four categories of invention and therefore claims 25 and 27-32 are not statutory. Energy is not a series of steps and thus is not a process. Energy is not a physical article or object and such is not a machine or manufacture. Energy is not a combination of substances and therefore not a composition of matter. Examiner made the recommendation that modifying the claim language to include “program product stored on a non-transitory storage medium” will overcome the rejection.

Argument EE: (Brief at 59)

Applicant argues that the term "storage" by itself excludes signals per se. (Brief at 59)

Response to Argument EE:

Examiner respectfully disagrees. A signal carries information from Point A to Point B. Said information is sent from Point A, **on** the signal, and received at Point B where it is used and/or stored. The fact that the information sent from Point A is

Art Unit: 2456

received and understood at Point B necessarily means that the information is ***stored*** on said signal, even if it dissipates after receipt. Otherwise, how could Point B possibly know what information point A sent? The information is not random, but purposefully sent. It carries information, which means it stores said information. Therefore, storage medium can be a signal, and since signals are non-statutory, Claims 25 and 27-32 are rejected under 35 U.S.C. 101.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/RICHARD G KEEHN/

Examiner, Art Unit 2456

Conferees:

/Michael Won/

Primary Examiner

AU 2456

/Rupal D. Dharia/

Supervisory Patent Examiner, Art Unit 2456